

Syst ms, methods, and apparatus for automatic updating of a programmer for an active implantable medical device

Patent Number: ☐ US5800473
 Publication date: 1998-09-01
 Inventor(s): FAISANDIER YVES (FR)
 Applicant(s):: ELA MEDICAL SA (FR)
 Requested Patent: ☐ JP10069388
 Application Number: US19970797379 19970207
 Priority Number(s): FR19960001564 19960208
 IPC Classification: A61N1/08
 EC Classification: A61N1/372B, G05B19/042P, G06F17/00D4B, G06F17/00D4C, G06F19/00A1
 Equivalents: DE69702273D, DE69702273T, ☐ EP0791373, B1, ☐ FR2744817

Abstract

A system and methods for the automatic update of the software of an external programmer implant is that is used to program and configure an active implantable medical device implant and acquire data obtained by the implant. The programmer comprises software composed of an assembly of software objects. The implant comprises a memory containing parametric data for the functioning of the implant and an assembly of software objects necessary for the functioning of the programmer in connection with the aforementioned of parametric data. The programmer and the implant communicate bi-directionally. The automatic updating preferably occurs by the programmer reading of the memory of the implant, establishing a list of objects that are found in the implant with their respective versions, comparing the list to the objects (and their versions) that are in the programmer software, downloading from the implant of those objects which are not found in the programmer software and adding these objects to the programmer software; and/or downloading from the implant those objects in the programmer software whose version is prior to the version of the object found in the implant, and to replace the programmer software objects, with the more current version.

Data supplied from the esp@cenet database - 12

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-69388

(43) 公開日 平成10年(1998) 3月10日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/445 19/00			G 0 6 F 9/06 15/42	4 2 0 M Z

審査請求 未請求 請求項の数18 O L 外国語出願 (全 33 頁)

(21) 出願番号	特願平9-25642	(71) 出願人	594199522 エエルア メディカル ソシエテ アノニ ム フランス国、セデクス、92541 モンルー ジュ、リュウ モーリス アルヌー 98- 100
(22) 出願日	平成9年(1997) 2月7日	(72) 発明者	イブ フザンディエ フランス国、75014 パリ、プールパール ラスバイユ 243
(31) 優先権主張番号	9 6 0 1 5 6 4	(74) 代理人	弁理士 浜田 治雄
(32) 優先日	1996年2月8日		
(33) 優先権主張国	フランス (F R)		

(54) 【発明の名称】 能動植込み型医療装置用プログラマの自動更新システム

(57) 【要約】 (修正有)

【課題】 能動植込み型医療装置得られたデータを収得する植込み装置用外部プログラマのソフトウェアを自動的に更新する。

【解決手段】 植込み装置は、動作についてのパラメトリック・データを記録するためのメモリと、前記データと結合してプログラマが動作するのに必要なソフトウェアアセンブリとを備える。プログラマと植込み装置は双方向に交信する。自動的な更新は、好適には、プログラマが植込み装置のメモリを読み取り、植込み装置内において検出された各オブジェクトおよびそれらのバージョンのリストを形成し、このリストをプログラマ・ソフトウェア内のリスト (ならびにバージョン) と比較し、プログラマ・ソフトウェア内においては検出されなかったオブジェクトをインプラントからダウンロードする。

【特許請求の範囲】

【請求項 1】 能動植込み型医療装置の植込み装置と、植込み装置外部プログラマとからなり；プログラマはさらに少なくとも一つのソフトウェア・オブジェクトを有するソフトウェアを備え；植込み装置はさらにこの植込み装置の機能に相関するパラメトリック・データを記録するメモリとプログラマが前記パラメトリック・データに結合して機能するために必要とされる少なくとも一つのソフトウェア・オブジェクトとを備え、プログラマおよび植込み装置はさらにこれらの間を双方向に接続して情報交換を行うための手段と、前記少なくとも一つのソフトウェア・オブジェクトの少なくとも一部を植込み装置からプログラマへ選択的に伝送するための手段とを備え；プログラマは植込み装置のメモリ内に含まれた前記少なくとも一つのソフトウェア・オブジェクトの少なくとも一部を植込み装置からダウンロードすることを指令するよう動作し得るとともに、ダウンロードされた前記ソフトウェア・オブジェクトの一部をプログラマに追加および／またはこれによって代替することからなるシステム。

【請求項 2】 プログラマがさらに：読み込まれたソフトウェア・オブジェクトのリストを形成するための手段と；前記形成されたリストとプログラマ内のソフトウェアとを比較するための手段と；前記形成されたリスト内に存在し前記プログラマ・ソフトウェア内には存在しないオブジェクトを欠如するオブジェクトとして確認する手段と；確認された欠如するオブジェクトを植込み装置からダウンロードするとともに、この欠如するオブジェクトをプログラマ・ソフトウェアに追加する手段とからなる請求項 1 記載のシステム。

【請求項 3】 各植込み装置のソフトウェア・オブジェクトが日付に相当する結合されたバージョン・コードを有し、プログラマがさらに：植込み装置のメモリを読み取るとともに、植込み装置ソフトウェア・オブジェクトならびに結合されたバージョン・コードのリストを形成する手段と；形成されたリストをプログラマのソフトウェア・オブジェクトと比較するとともに、対応するプログラマ・ソフトウェア・オブジェクトのものより新しいバージョン・コードを有する植込み装置ソフトウェア・オブジェクトをより新しいソフトウェア・オブジェクトとして確認し；確認されたより新しいソフトウェア・オブジェクトを植込み装置からダウンロードするとともに、対応するプログラマ・ソフトウェア・オブジェクトを前記ダウンロードされたより新しいソフトウェア・オブジェクトによって代替する手段とからなる請求項 1 記載のシステム。

【請求項 4】 プログラマがさらに前記形成されたリストをプログラマ・ソフトウェア・オブジェクトと比較するとともに、前記形成されたリスト内に存在するが前記プログラマ・ソフトウェア内には存在しないオブジェ

クトを欠如するソフトウェア・オブジェクトとして確認し、

ダウンロード手段がさらに前記欠如するソフトウェア・オブジェクトをプログラマ・ソフトウェアに追加する手段を備える請求項 3 記載のシステム。

【請求項 5】 各植込み装置・ソフトウェア・オブジェクトがさらにオブジェクト確認、バージョン確認ならびにプログラム指令のブロックを備える請求項 1 記載のシステム。

10 【請求項 6】 前記少なくとも一つの植込み装置ソフトウェア・オブジェクトがさらに心臓ペースメーカの公称周波数を表記するためのオブジェクトを備える請求項 1 記載のシステム。

【請求項 7】 前記少なくとも一つの植込み装置ソフトウェア・オブジェクトがさらに心臓ペースメーカの時間にしたがって測定した周波数を表記するためのオブジェクトを備える請求項 1 記載のシステム。

【請求項 8】 メモリと、

前記メモリ内のソフトウェア・プログラムと、

20 植込み装置が機能するためのパラメトリック・データとからなるあらかじめ設定された機能を備え、ソフトウェアがさらにプログラマがパラメトリック・データと結合して動作するための少なくとも一つのソフトウェア・オブジェクトを備え、前記少なくとも一つのオブジェクトを外部プログラマにダウンロードすることができる能動植込み型医療装置の植込み装置。

【請求項 9】 各ソフトウェア・オブジェクトがさらに日付に相当する結合されたバージョン・コードを備える請求項 8 記載の植込み装置。

30 【請求項 10】 ソフトウェアがさらに前記植込み装置内におけるダウンロード可能な各ソフトウェア・オブジェクトのリストを備える請求項 8 記載の植込み装置。

【請求項 11】 各ソフトウェア・オブジェクトがさらに日付に相当する結合されたバージョン・コードを備え、前記リストがさらに各ソフトウェア・オブジェクトについて前記バージョン・コードを備える請求項 10 記載の植込み装置。

【請求項 12】 能動植込み型医療装置の植込み装置に適用するための外部プログラマであり、：少なくとも一つのソフトウェア・オブジェクトを有するソフトウェアと；植込み装置の機能のデータとプログラマが前記植込み装置と結合して動作するために必要なソフトウェア・オブジェクトとを植込み装置からプログラマへ選択的に転送することによる双方向の情報交換接続と；前記少なくとも一つのソフトウェア・オブジェクトの少なくとも一部を植込み装置からダウンロードし、前記ダウンロードされたソフトウェア・オブジェクトをプログラマのオブジェクトに追加および／またはこれによって代替する手段とからなる外部プログラマ。

50 【請求項 13】 ダウンロード手段がさらに：植込み装

置のメモリを読み取り；前記植込み装置メモリ内において検出されたオブジェクトおよびそれに対応するバージョン・コードのリストを形成し；形成されたリストを少なくとも一つのプログラマ・ソフトウェアのオブジェクトと比較し；プログラマ・ソフトウェアの欠落するオブジェクトを植込み装置からダウンロードするとともに、このオブジェクトをプログラマ・ソフトウェアに追加するよう指令し、および／またはプログラマ・ソフトウェア内に存在するが植込み装置内で検出されたものより以前のバージョンを有するオブジェクトを植込み装置から

10

ダウンロードし、これによってプログラマ・ソフトウェア内のオブジェクトを代替するよう指令することからなる請求項 12 記載のプログラマ。

【請求項 14】 各ソフトウェア・オブジェクトがさらに少なくとも一つのオブジェクト確認、バージョン確認、およびプログラム指令のブロックを備える請求項 12 記載のプログラマ。

【請求項 15】 a) オブジェクトに基づいたソフトウェアを有するプログラマならびに前記プログラマ内のオブジェクトのリストを設け；

b) オブジェクトに基づいたソフトウェアと、複数のオブジェクトと、植込み装置内のオブジェクトのリストとを有する植込み装置を設け；

c) プログラマを前記植込み装置と交信するように配置し；その後自動的に、

i) 前記植込み装置のメモリ内のオブジェクトのリストを読み込み；

i i) 前記植込み装置のメモリ内のオブジェクトのリストを前記プログラマ内のオブジェクトのリストと比較し；

i i i) 前記プログラマ内のオブジェクトならびに前記プログラマ内の前記リストを更新することからなる植込み装置用プログラマの更新方法。

【請求項 16】 ステップ (c) (i i) がさらに前記植込み装置のリスト内に存在するが前記プログラマのリスト内には存在しない各オブジェクトを確認することからなり、ステップ (c) (i i i) がさらに前記確認された各オブジェクトを前記プログラマにダウンロードし、前記プログラマ内のオブジェクトのリストを修正することからなる請求項 15 記載の方法。

【請求項 17】 ステップ (a) がさらに各プログラマ・オブジェクトをバージョン・コードと共に設けることからなり；ステップ (b) がさらに各植込み装置オブジェクトをバージョン・コードと共に設けることからなり；ステップ (c) (i) がさらに各植込み装置オブジェクトのバージョン・コードを読み取ることからなり；ステップ (c) (i i) がさらに読み取った植込み装置オブジェクトのリスト上の各オブジェクトのバージョン・コードをプログラマ・オブジェクトのリスト上のオブジェクトのバージョン・コードと比較するとともに、各

植込み装置オブジェクトについて、そのバージョン・コードが対応するプログラマ・オブジェクトのバージョン・コードより新しいかどうかを判断することからなり；ステップ (c) (i i i) がさらに判定された新しい各植込み装置オブジェクトおよびそのバージョン・コードをダウンロードするとともに、前記対応するプログラマ・オブジェクトおよびそのバージョンコードを前記ダウンロードされたオブジェクトおよびそのバージョン・コードによって代替することからなる請求項 16 記載の方法。

【請求項 18】 ステップ (c) (i i) がさらに前記植込み装置のリスト内に存在するが前記プログラマのリスト内には存在しない各オブジェクトを確認することからなり、ステップ (c) (i i i) がさらに前記確認された各オブジェクトを前記プログラマ内にダウンロードするとともに、前記プログラマ内のオブジェクトのリストを修正することからなる請求項 17 記載の方法。

【発明の詳細な説明】

【0001】

20 【発明の分野】この発明は、能動植込み型医療装置と交信しデータを交換するための外部プログラミング機器に関する。

【0002】

【発明の背景】この発明は主に心臓ペースメーカのケースについて記述するが、これは本発明の実施の一例にすぎず、本発明は一般的な方式において非常に多様な“能動植込み型医療用装置”に適用することができる。能動植込み型医療用装置は、例えば 1990 年 6 月 20 日付の EC 評議会の指令第 90/385/CEE 号により定義されており、心臓ペースメーカに加えて細動除去器および／または電気的除細動器、医療用剤の拡散／静脈注入ポンプ、移植蝸牛刺激装置、植込み型生理学的コレクタ等を含むものである。

【0003】これらの患者に適用、すなわち植え込まれる機器（以降選択的に“植込み装置”と称する）は、所要の機能を適宜な方法で実行するために、外部プログラミング装置によってプログラムされる（または特にある程度まで再プログラム可能なもの等）。外部プログラム装置は主に“プログラマ”と呼ばれ、一般的に植込み装置部位の近くに配置されたプログラミング・ヘッドを備えるマイクロコンピュータである。プログラマは、電磁伝送手段（ワイヤレス遠隔測定）により、植込み装置内に蓄積されたデータをプログラミング・ヘッドを介して読み込むことができる。例えば、プログラマは、パラメトリック・データ、植込み装置によって記録された生理信号、植込み装置によって診断された事象等を読み込むことができる。さらに、プログラマは、特に植込み装置を再プログラムするためにパラメトリック・データをこれに伝送することができ、これはすなわち植込み装置の機能の幾つかのパラメータを変更するものであり、また

40

50

場合によって植込み装置に対してプログラミング指令を伝送してそのソフトウェアを再構成することができる。ここで“パラメトリック・データ”という用語は植込み装置の機能をプログラムおよび／または制御するために使用するパラメータを意味するとともにこれを含むものであると理解すべきであり、これらのパラメータは植込み装置の構成時においてあらかじめ設定するか、または装置の意図する使用に応じてプログラムされた／プログラム可能な数値とすることができ、ここでは“データ”と呼称する。

【0004】既存のプログラマは所与のタイプの植込み装置、または同じ製造元の一連の植込み装置に対して特有のものである。したがって、一般的に、新しい植込み装置が市場の配給される度にこれに対応して適用可能なプログラマの更新を提供する必要がある。加えて、新しい植込み装置の市場への投入時以外でも、プログラマ・ソフトの更新を必要とする改良が常時なされている。

【0005】この更新は非常にデリケートなものである。実際に、明確な安全上の理由のため、全ての更新は非常に厳密な規則にしたがって付与されなければならない：規則はプログラマが保管されている所定のヘルスセンタまたはクリニックを訪問することを要求しており、それによって古いプログラマが有資格者によって変更および修復される。この訪問は種々の確認文書の記入および署名等をもって実行される。この手続きは面倒かつコストを要するものである。結果として、あまり頻繁にこれを遂行することは避けるものである（例えば、ソフトウェアを僅かに変更した場合に更新を行わないことがあり得る）。

【0006】加えて、安全上の手続きがどれほど完璧であっても、適正な更新が欠如する危険性は完全には排除されない。例えば、製造元を確認することができないプログラマは更新することができない。また、前記の手続きを行うことによって、更新転換中の動作、また特に不適格な旧バージョンのソフトの適用の危険性を防止することはできない。“旧”という表現は、単に更新の実施によって古くなった先のバージョンを意味するものと理解すべきである。

【0007】

【発明の目的および概要】したがって、本発明の目的の一つは、人間が介在することなく完全に自動的な方式によってソフトウェアを更新する新しいコンセプトのプログラマを提供することによりこれらの難点を克服することであり、これによって現在の更新処理に関する面倒、コスト、および危険を排除または削減する。

【0008】本発明は、ソフトウェアの概念、すなわち広範な概念的構成要素（データ、構築、分解および方法）における“オブジェクト”の概念の実施に基づいており、ソフトウェアがデータを非常に明確な構造をもって処理および表示することを可能にする。

【0009】特に、データへのアクセスは一般的にオブジェクトにしたがった方法によってのみ許可される。このことは、オブジェクトのデータがあらかじめ設定された範囲を超過することを防止し、ソフトウェアに高い信頼性を提供することを可能にする。

【0010】さらに、“遺産”または“内在”のコンセプトが、一つのオブジェクトから他のオブジェクトへの継承を行うソフトウェアの設計を容易にする。このことは、以下に説明するように、一つのオブジェクトが他のオブジェクトの方式を直接的に使用することを可能にし、またオブジェクトの変更または追加による更新に関してそれぞれの構造を簡略化することができる。この種のソフトウェアのオブジェクトおよびその心臓ベースメーカの分野における適用の実例について以下に説明する。

【0011】一般的に、このオブジェクト・プログラミングの実施において、例えばマックス・ブーシェ著“ラ・ドマルシェ・オブジェーコンセプト・エト・オウティル” AFNOR, 1994年発刊（ザ・ステップ・オブジェクト・コンセプトおよびツール, AFNOR, 1994年発刊）等の公開文書を参照することができ、また同様に例えばC++等の“オブジェクトにしたがった”典型的なプログラミング言語の資料を参照することができ。

【0012】一般的に、本発明は（i）植込み装置内にオブジェクトに応じたソフトウェアを記録し、そのデータの表示、処理、およびプログラミングを外部コンピュータによって設定することを可能にし；（ii）プログラマ内においてこのプログラマが内包するオブジェクトのリストと植込み装置が内包するオブジェクトのリストを比較し；さらに（iii）植込み装置内のオブジェクトのうちプログラマ内に存在しないものあるいはプログラマ内のオブジェクトのバージョンに比してより新しいものを植込み装置からプログラマへダウンロードし、これによってプログラマのソフトウェアを更新することからなる。

【0013】以下により詳細に説明するが、この方法によれば従来の更新プロセスに比べて少なくとも4つの大きな利点を提供することができ：それらは安全性、経済性、プログラミングの簡略化、ならびに規格化である。

【0014】安全性

全てのプログラマは最低限の基本ソフトを備えており、これはすなわち、植込み装置との更新の実行を可能にするオブジェクト、および植込み装置内のオブジェクトとプログラマ内のオブジェクトとの間の接続の処理を可能にするオブジェクトである。したがって、この種の基本プログラマは、植込み装置がプログラマより後に開発されプログラマ内には含まれていない内容を含んでいる場合においても、各種の植込み装置ならびのそのオブジェクトを読み取ることができる。その結果、本発明によれ

ば、プログラムの操作者は、いかに古くとも互換性のあるプログラマを所有していれば、最新の植込み装置モデルの機能の読み取りおよび再プログラムが不可能となることはない。加えて、各パラメータは植込み装置のソフトウェアによって定義された適正な処理モードを有し、これがプログラマによるこれらのパラメータの捕捉を制御し、これは所与のオブジェクトにおいて定義される。したがって、プログラマが誤ったパラメータ数値を捕捉することはなく、これはこれがプログラマ内にあらかじめ存在したオブジェクトではなく植込み装置によって発行されたオブジェクトによって制御されるからである。

【0015】経済性

プログラマの更新に要する費用は大幅に縮小され、植込み装置の仕様変更（すなわち、新しい世代のプログラマ、ハードウェアが遠隔測定ヘッドを交換した際等）に際してのみ必要とされるようになる。既存の更新変更によって生じる従来の安全手続きの費用も全て削除することができ、すなわち各プログラマ設置場所に対する有資格者の派遣、更新確認登録の署名、古いディスクットのバックアップ等を省略することができる。

【0016】プログラミングの簡略化

プログラマ・ソフトウェアの進化および更新は、プログラマの使用者にとって明白な方式によって自動的に実施される。

【0017】基本的なオブジェクトの処理を実行するためにごく一般的な機能のみが必要とされ、付加されたオブジェクトのリストは、新しい植込み装置内において定義された新規または追加的なオブジェクトによって最後に実行される。

【0018】規格化

本発明の実施の柔軟性は、全てのプログラマが各植込み装置をその製造元および型式にかかわらず互換的に読み取ることを可能にする規格の提供を可能にする。各メーカーはプログラマのグラフィック表示特性、可能なデータ処理、補足的計算等を所定のオリジナリティおよび製品の差別性を維持する方法によって定義することができるが、基本的なオブジェクトは同一の方式で定義され、したがって“異種の”植込み装置の読み込みが完全に可能になる。

【0019】したがって、ユニバーサル・プログラマの実現をもくろむことが可能になるが、これは従来はコストおよび複雑性の理由のみならず、安全性の理由（すなわち予知されない非互換性のリスクによる損害）からも除外されていた。

【0020】本発明の典型的な一側面は、一方で能動植込み式医療装置型の植込み装置を有し、他方で外部プログラマを有するシステムからなり、これにおいて：プログラマは複数のソフトウェア・オブジェクトからなるソフトウェアを有し；植込み装置は、一方で植込み装置の機能に関するパラメトリック・データを含むとともに、

他方で前記植込み装置のパラメトリック・データと結合したプログラマの動作のために必要とされる複数のソフトウェア・オブジェクトを含むメモリを有し；情報を交換するためにプログラマと植込み装置とを双方向的に接続し、特に前記パラメトリック・データの全てまたは一部ならびに前記植込み装置の一つまたは複数のオブジェクトをプログラマへ選択的に伝送することを可能にし；プログラマは、植込み装置メモリ内に記録された前記オブジェクトの少なくとも一部を植込み装置からダウンロードし、これらのオブジェクトをプログラマソフトウェア内に定義されたオブジェクトに追加および／または代替するよう指令するためのプログラミング能力を有する。

【0021】より好適には、植込み装置との完全な互換性を達成するためにできる必要なぎり多くまたは多数の植込み装置オブジェクトをダウンロードする。結果として、システムがプログラマを更新することができる。好適には、プログラマが植込み装置と交信される度に更新処理が自動的に実施される。

【0022】好適には、プログラマ更新能力の動作は：植込み装置内において検出されたオブジェクトのリスト、より好適には検出されたオブジェクトのバージョンを含むリストを形成し；形成されたリストをプログラマソフトウェア内のオブジェクトのリストと比較し；プログラマソフトウェア内に存在しないオブジェクトを植込み装置からダウンロードし、これらのオブジェクトをプログラマ・ソフトウェアに追加するよう指令し；および／またはプログラマ・ソフトウェア内に存在するもののバージョンが植込み装置内で検出されたものより以前のバージョンであるオブジェクトを植込み装置からダウンロードし、これによってソフトウェア内のオブジェクトを代替するよう指令することからなる。

【0023】結果として、更新されたプログラマが次に同様な植込み装置と交信する際、植込み装置内のオブジェクトとプログラマとの比較が全ての最新のバージョンのオブジェクトをプログラマ内に検出し、ダウンロードは必要とされない。

【0024】本発明の他の側面は、独立した製品に応じて、一方で植込み装置に対して、他方ではプログラマに対して、それぞれ前述した特性のうち適切なものが実施される。

【0025】

【発明の詳細説明】次に、本発明の好適な実施例に基づいた実例について説明する。

【0026】前述したように、プログラマならびに植込み装置のソフトウェアは、オブジェクトに基づいたソフトウェア“オブジェクト”において組織される。

【0027】心臓ペースメーカーからなる植込み装置の場合、ペースメーカーの各要素（パラメータ）を一つのオブジェクトによって現し、これはプログラム可能である

か、または単にプログラムのスクリーン上に表示あるいはプリンタ上に印刷可能である。

【0028】したがって、心臓周波数を時間の関数として与える曲線は、特に“心臓周波数”と呼ばれるソフトウェア・オブジェクトによって現され、これはより一般的なオブジェクトから派生し、さらにこれは“曲線”と呼ばれるとともにそのフォーマット、タイトル、表示方法、印刷方法等においてある曲線に相応する一連のデータを内包する。ここで“フォーマット”という概念は、横座標（X-軸）および縦座標（Y-軸）のスケールに 10 関するものである。

【0029】プログラムはその一部分に対してオブジェクト“接続管理者”を有し、これは必要に応じて圧縮／拡大等の送信データプロトコルを適用することによりベースメカ内に記録された心臓周波数に相当するデータを呼び出すことができる。続いて、プログラムはプログラマ・スクリーン上にオブジェクト曲線を使用して心臓周波数データを現すウィンドウを表示し、ここで拡大率、ウィンドウ内における移動、カーソルを使用しての表示されたデータの検出等の考えられる全ての表示機能を可能にする。これらの機能は一般的な方法によって実施可能であることが理解される。 20

【0030】本発明のコンセプトは3原則の仮説に基づいている。第一に、プログラマ・ソフトウェアは“オープン”であり、すなわちデータ、データの表示、データの処理、および植込み装置のプログラミングに関する新規のオブジェクトの受信および組み込みにより変更することができる。これらのオブジェクトは、例えば接続管理オブジェクトの予備的なローディングにより、（本発明の特徴にしたがって）植込み装置、ディスクまたは他の既知のコンピュータ補助装置（例えばROM交換、モデムまたはファイル転送等）からもたらされる。 30

【0031】第二に、植込み装置はプログラマ表示、データ処理、ならびに植込み装置内に含まれるまたはこれによって必要とされるデータのプログラミングを定義するために必要とする全てのオブジェクトを蓄積する。植込み装置ソフトウェア内のオブジェクトのサイズは、既にプログラマ内に存在する他のオブジェクトから継承されるものの大きさに比例して縮小されることを理解する必要がある。例えば、プログラマ・オブジェクト曲線は、一般的に曲線の表示および処理を定義し、したがって植込み装置ソフトウェアのオブジェクト心臓周波数はオブジェクト曲線に表示されるデータの固有の特徴のマップのみを必要とする。したがって、オブジェクト曲線は 40 実際の処理および／または表示を管理する。

【0032】第三に、オブジェクトのバージョンは日付を付けられており（例えば、バージョン番号および／または日付コードによって）、プログラマ内におけるいずれかのより新しいバージョンが以前のものを代替する。したがって、オブジェクトの開発は、“情報”互換性を 50

付加することによって実施され、すなわち最新のバージョンは古いバージョンに関する全てのデータ変更に対する互換性を維持する。

【0033】植込み装置とプログラマとの間のデータ交換の開始時において、プログラマはオブジェクトのリストと呼ばれるブロックを体系的に呼び出し、これは植込み装置内における可能なオブジェクトのレパートリである。これは、好適には、交信接続が達成されると同時に自動的に実施することができる。このオブジェクトのリストは、植込み装置内に保持される。好適には、オブジェクトのリストは、さらに植込み装置内の各オブジェクトのバージョンを含んでいる。さらに好適には、オブジェクトのリストは、オブジェクト番号およびオブジェクト名（さらに選択的にその説明）からなる表を含む。

【0034】続いて、プログラマは、植込み装置オブジェクトを既知のオブジェクト（すなわち、プログラマソフトウェア内の可能なオブジェクトのリストで、これは動的に管理されるときに構成ファイル内において保護される）と比較する。これらのオブジェクトのうちの一つまたは複数がプログラマにとって未知である場合、プログラマはオブジェクト接続管理者を呼び出し、この未知のオブジェクトのダウンロード処理を開始し、これによってそのソフトウェアを完全化するとともにプログラマを植込み装置に適応させる。プログラマのコンフィグレーションファイルはその時更新される。

【0035】ダウンロードおよび更新処理は、プログラマがダウンロードされるオブジェクトを継承する接続管理オブジェクトを使用することにより、未知の各オブジェクトに対して自動的に実行される。接続管理者は例えばオブジェクト数等のコードをベースメカに送信し、ベースメカは対応する情報のブロックをプログラマに返送する。

【0036】プログラマのソフトウェア内に存在するが植込み装置内に含まれるものより古いバージョンに相当するオブジェクトに対して、同様な処理が実行される。

【0037】この更新局面の後に、プログラマは植込み装置に互換するために必要とする全てのオブジェクトを有し、植込み装置内に含まれる全てのデータを呼び出し、所要の結果を表示、植込み装置を再プログラム等を既存の方式において実施することができる。

【0038】したがって、オブジェクト心臓周波数を有しこのオブジェクトを認識していないプログラマによって読み取られる植込み装置の場合において、プログラマはベースメカによって提供されるオブジェクトのリスト内のオブジェクトを確認する。続いて、プログラマは植込み装置にオブジェクトを送送するよう要求し、オブジェクトはこれを実行し、プログラマはこれをプログラムに追加する。オブジェクトの構成者へのコールが、後にオブジェクトに関連付けられた周波数曲線、ディスプレイ・ウィンドウの選択、および各種変数の初期化を受

信するためのメモリ・ゾーンを確保することを可能にする。

【0039】次に、オブジェクト心臓周波数のコード（または番号）が伝送され、これがプログラムのソフトウェア内に維持されている可能なオブジェクトの動的リスト内に付加される。

【0040】プログラムのオペレータが曲線を表示することを要望する際、プログラムに所要のウィンドウを要求する。そこで、プログラムはメモリ内における曲線の存在を検査し、曲線が存在しない場合、ペースメーカ内に記憶された心臓周波数データの読み取りを開始する。その後、曲線が一般的なディスプレイ調節制御（ゲイン、ズーム、動作）、メジャー（カーソル）を有するウィンドウ内に表示され、可能的には、既にプログラム内に存在する一般的オブジェクト曲線によって全て制御されるこれら各種の機能をプリントする。

【0041】次に、ペースメーカ・オブジェクトの二つの詳細な例について説明する：これらはオブジェクト公称周波数（プログラムされた固定値）；およびオブジェクト呼吸周波数（時系的に変化するデータ）である。

【0042】オブジェクトの表記は幾つかの形式において実施することができ、これらは3つのケースにまとめることができ：それらは、（1）機械コード、（2）プリコンパイルされたコード、（3）超言語であり：機械コードまたはファームウェアは長期的な使用を可能にするためにプロセッサに結合される。コンパイルされたコード（例えば旧式の“P-コード”等）は非常に効果的であり、新規のオブジェクトの全てのデータおよび方式を表記するのを可能にする。しかしながら、これは大きなサイズを維持し、そのため小さな記憶容量を有する植込み装置への使用にはあまり適していない。超言語はプログラムによってあらかじめ設定された元のオブジェクトに属する変数値のみを表記する。これによれば、データ量が他のケースに比して大幅に縮小される。

【0043】植込み装置のデータの殆どは、超言語形式の表記で蓄積されるが、プリコンパイルされたコードへの要求は、元のオブジェクトとの合成のためアクセシブルを維持する必要がある。

【0044】ここに記述されている例においては、超言語が選択されており、これは縮小されたオブジェクトのサイズおよびプログラム言語からの完全な独立を提供する。他方で、プログラムの基本的オブジェクトへの対比のみを可能にし、これらを直接進展させることは不可能である。

【0045】プログラムが全く新しいコンセプトを統合することができない際にのみその限界に到達し、その結果、従来のプログラムのソフトウェアの更新に付随するこの種の現象のごく頻繁な発生を防止するために、基本的オブジェクトの開発が非常に良好に達成される。

【0046】例1

オブジェクト：公称周波数

公称周波数（基本周波数としても知られている）は全ての心臓ペースメーカの本質的なパラメータであり、以下の方法においてオブジェクトによって表記される：

【0047】1. オブジェクト番号：例えば8文字（製造元について2文字、オブジェクトについて4文字、バージョン番号について2数字）からなる固有のコードがオブジェクトに関連付けられる。このことにより、固有であるとともに容易に判読できるコードを選択することが可能になる。

【0048】2. オブジェクト祖先：このオブジェクトに対する番号がオブジェクト祖先を提供し、これはこの場合においてプログラムの内に既に存在する唯一のオブジェクトであり、数値の表示、捕捉、印刷および記憶の全ての処理をプログラムが管理することを可能にする。このオブジェクトは、当然、最小限のより一般的なオブジェクトを継承し、それは接続管理器である。これはあらかじめ設定されたステップによるプログラミングに係るため、数値はステップの数内においてコード化される。このデータは5つの文字からなる。

【0049】次のデータがオブジェクト祖先のパラメータを提供する：

(a) 表示に際して数値を伴う記述的文章（100文字）（例えば、20文字のテキストが、それぞれ20文字からなる5つの異なった言語による合計100文字に対応する）

(b) メジャーの単位（10文字）

(c) 最小、最大および公称数値（16文字）：オブジェクト祖先内に限界値が存在し、これによって限界値を超えた数値の捕捉を防止する。これらの限界値は、“公称値の設定”の方法の呼び出し中にオブジェクト内に以前に存在した数値を最大値、最小値および公称値によって置き換えることによって更新され、これらは全てのプログラム可能なオブジェクトを継承する。ペースメーカ内に存在する多様なケースに適応するために、ステップの増進を定数、対数または逆関数値として定義することができる。

(d) 参照ウィンドウ（5文字）：プログラムのスクリーンの一つにおいて表示される各データであり、一つまたは複数のウィンドウを含むことができる。各ウィンドウはペースメーカによって少なくとも一つのタイトル、形式（1列、2列等）、および実行可能な動作（移動、現時点または継続的なペースメーカの読み取り、ローカル・メニュー等）を用いて定義される。

(e) 参照ウィンドウ内における位置（5文字）：これは5数字までによってまとめることができる列、高さ等を提供し：最初のものが列を所与し、続く3つのものが相互の関係における優先順位を所与し；スクリーン制御の核がこれらを優先順位にしたがって上から下に配置する。

【0050】したがって、このオブジェクトに対するデータ全体は単一言語においては56文字、複数言語においては136文字を有する。

【0051】例2

オブジェクト：呼吸周波数

呼吸周波数を解析するペースメーカの場合において、得られたデータは数時間、数日および数カ月の期間で記憶される。

【0052】オブジェクトは以下の方式にしたがって超言語で表記することができる（この概念は既に説明したので必要なもの以外は反復しない）：

(1) オブジェクトの番号 (8文字)

(2) オブジェクト祖先 (5文字)：これは時間 $f(t)$ の関数としての周波数発生曲線に関し、垂直および水平スケール、データのフォーマット、表示モード（ライン、最小-最大等）、スケール変更方法、表示および印刷を有する。

【0053】次のデータがオブジェクト祖先のパラメータを提供する：

(a) 記述的文章 (各言語について20文字、または5言語による複数言語モードにおいて100文字)：

(b) 測定単位 (10文字)

(c) データのフォーマット (5文字)：これは数値（正数）の連続（8, 16, 24, または32ビット）、実数等からなる。このフォーマットは存在するオブジェクトであり、オブジェクト接続管理者を継承し、これがペースメーカとプログラマとの間におけるデータの交換を可能にする。

(d) 各データを分離する時間周期 (4文字)：これは4バイトの単語によって符号化されたミリ秒単位の時間周期であり、時間の変動範囲は1msないし1137時間となる。

(e) 垂直スケール表記 (12文字)：これは低値、高値、ステップ増加数からなる。

(f) 参照ウィンドウ (5文字)：公称周波数の場合と同様である。

(g) 参照ウィンドウ内における位置 (5文字)：公称周波数に対するものと同様に、グラフ位置の表示は所定の優先順位にしたがったものとなり；列の数および表示される要素の数がウィンドウによって計数され、これによって可能な各最大位置内に収容され：したがって、一つの曲線のみを含む1列を有するウィンドウはこれを可能な面の全ての上に表示し、2つの列および4つの曲線を有するウィンドウはそれぞれを全ての面、領域等の1/4の上に表示する。

【0054】したがって、これらのデータ全体は、単一言語モードにおいては74文字、5言語からなる複数言語モードにおいては154文字を有する。

【0055】したがって、ペースメーカの動作に対する全てのパラメータを同一の総合的方式によって表記することができる。

【0056】比較的複雑なペースメーカの場合においては、例えばメモリ内に記録された20個の曲線と50個のプログラム可能なパラメータからなり、したがって、オブジェクトの全体的なサイズは複数言語モードにおいて、パラメータに対して $50 \times 136 = 6800$ 文字、曲線に対しては $20 \times 154 = 3080$ 文字、全体では9880文字となる。さらに、更新を実行する前にプログラムに伝送するオブジェクトのリストのブロックを追加する必要があり、これは $70 \times 8 = 540$ バイトからなる。

【0057】したがって、ペースメーカ内のオブジェクトの蓄積のために確保される全体的なサイズは、大抵の複雑なもの（例えば、最新型のDDDレート応答型ペースメーカ）の場合において約10kbに達する。好適には、これらの要素はペースメーカのROM回路内に過剰な消費を要することなく容易に入力することができ、したがってこの回路は能動化されない。

【0058】当業者においては、本発明が記載された実施例以外によっても実施可能であり、これは単に説明の目的のものであり、これに限定されるものではないことが理解されよう。

【外国語明細書】

1. Title of Invention

AUTOMATIC UPDATING OF A PROGRAMMER FOR AN
ACTIVE IMPLANTABLE MEDICAL DEVICE

2. Claims

1. A system comprising, an active implantable medical device implant, and an external implant programmer, wherein:

the programmer further comprises a software having at least one software object;

the implant further comprises a memory containing parametric data associated with the functioning of the implant and at least one software object necessary for the functioning of the programmer in connection with said parametric data, wherein the programmer and the implant further comprise means to establish between them a bi-directional connection to exchange information, and means for allowing the selective transfer of at least a part of said at least one software object from the implant to the programmer; and

the programmer is operable to command the downloading from the implant of at least a part of said at least one software object contained in the implant memory and to add and/or to substitute the downloaded part of said software object to the programmer.

2. The system of the claim 1, in which the programmer further comprises:

means for establishing a list of software objects read;

means for comparing said established list and the software objects of the software in the programmer;

means for identifying as an absent software object an object in said established list and not in said programmer software,

means for downloading from the implant of an identified absent object and adding the absent objects to the programmer software; and

3. The system of claim 1 which wherein each implant software object has an associated version code corresponding to a date, and the programmer further comprises:

means for reading the memory of the implant and establishing a list of implant software objects and associated version codes;

means for comparing the established list to the programmer software objects and identifying a newer software object as an implant software object having a version code that is newer than the corresponding programmer software object; and

means for downloading from the implant the identified newer software and replacing said downloaded newer software object for the corresponding programmer software object.

4. The system of claim 3 wherein the programmer further comprises means for comparing said established list to the programmer software objects and identifying an absent software object as an object in said established list and not in said programmer software, and

wherein the downloading means further comprises means for adding said absent software object to the programmer software.

5. The system of claim 1 wherein each implant software object further comprises an identifying object, an identifying version and a block of program instructions.

6. The system of claim 1 wherein said at least one implant software object further comprises an object to display a nominal

frequency for a cardiac pacemaker.

7. The system of claim 1 wherein said at least one implant software object further comprises an object to display a cardiac frequency over a time measure.

8. An active implantable medical device implant having a predetermined function comprising:

a memory,

a software program in said memory, and

parametric data for the functioning of the implant, wherein the software further comprises at least one software object for the functioning of a programmer in connection with the parametric data, said at least one object being downloadable to an external programmer.

9. The implant of claim 8 wherein each software object further comprises an associated version code corresponding to a date.

10. The implant of claim 8 wherein the software further comprises a list of each downloadable software object in said implant.

11. The implant of claim 10 wherein each software object further comprises an associated version code corresponding to a date, wherein said list further comprises said version code for each software object.

12. An external programmer for use with an active implantable medical device implant, comprising:

a software, having at least one software object;

bi-directional information exchange connection by selective transfer from the implant to the programmer, of a datum of the functioning of the implant and a software object necessary for the functioning of a programmer in connection with the aforementioned implant; and

means for downloading from the implant of at least a part of said at least one software object and to add and/or to substitute said downloaded software object to the objects of the programmer software.

13. The programmer of claim 12, in which the means for downloading further operates to:

read the memory of the implant;

establish a list of objects found in said implant memory with their respective version code;

compare the established list to at least one of the objects of the programmer software, and

command the downloading from the implant of absent objects of the programmer software and to add these objects to the programmer software, and/or

command the downloading from the implant of objects present in programmer software but whose version is prior to the one found in the implant, and to replace these objects in the programmer software.

14. The programmer of claim 12, in which each software object further comprises at least an identifying object, an identifying version and a block of program instructions.

15. A method for updating a programmer for an implant comprising:

- a) providing a programmer with object oriented software and a list of objects in said programmer;
- b) providing an implant with object oriented software, a plurality of objects, and a list of objects in said implant;
- c) placing the programmer in communication with said implant; and automatically thereafter
 - i) reading the list of objects in said implant memory;
 - ii) comparing the list of objects in said implant memory to said list of objects in said programmer; and
 - iii) updating the objects in said programmer and said list of objects in said programmer.

16. The method of claim 15 wherein step (c)(ii) further comprises identifying each object in said implant list that is not in said programmer list, and step (c)(iii) further comprises down-loading each said identified object into said programmer and modifying the list of objects in said programmer.

17. The method of claim 16 wherein:

step (a) further comprises providing each programmer object with a version code;

step (b) further comprises providing each implant object with a version code;

step (c) (i) further comprises reading the version code of each implant object;

step (c) (ii) further comprises comparing the version code of each object on the list of implant objects read to the version code of the objects on the list of programmer objects and determining, for each implant object, if its version code is newer than a corresponding programmer object version code; and

step (c) (iii) further comprises downloading each determined newer implant object and its version code and replacing said corresponding programmer object and version code with said downloaded object and version code.

18. The method of claim 17 wherein in step (c) (ii) further comprises identifying each object in said implant list that is not in said programmer list, and step (c) (iii) further comprises downloading each said identified object into said programmer and modifying the list of objects in said programmer.

3. Detailed Description of Invention

Field of the Invention

The present invention is directed to external programming machines that communicate with active implantable medical devices and exchange data therewith.

Background of the Invention

The invention will be mainly described in the case of a cardiac pacemaker, but this is only one example of an implementation of the present invention, which is applicable in a far more general manner to a very great variety of "active implantable medical devices." Active implantable medical devices are defined, for example, by the June 20, 1990 Directive 90/385/CCE of the European Community Council, to include, in addition to cardiac pacemakers, defibrillators and/or cardiovertors, diffusion/infusion pumps of medical substances, cochlear implants, implanted biological collectors, etc.

These devices (hereafter collectively designated "implants"), once put in place in a patient, that is, implanted, are programmed (and typically also reprogrammable to one extent or another) by an external programming device to perform the intended function in a desired manner. The external device is typically called a "programmer" and generally is a microcomputer

equipped with a programming head placed in proximity with the site of the implant. The programmer can read, by way of electromagnetic transmission (wireless telemetry) through the programming head the data stored in the implant. For example, the programmer can read the parametric data, biological signals that have been recorded by the implant, events diagnosed by the implant, etc. The programmer typically also is able to send parametric data to the implant so as to re-program it, that is to say to modify some parameters of implant functioning, and may be able to send programming instructions to the implant to reconfigure its software. The term "parametric data" should be understood to mean and include the parameters used to program and/or control the functioning of the implant, which parameters may be predetermined at the time of construction of the implant or programmed/programmable values based on the intended use of the device, and also are referred to herein as "data".

The known programmers are specific to a given type of implant, or to a family of implants of the same manufacture. Consequently, each time a new implant is placed on the market, it is generally required to provide a corresponding update of the applicable programmers. In addition, even in the absence of placement on the market of a new implant, improvements are regularly obtained that can require an update of the programmer software.

These updates are particularly delicate. Indeed, for self-evident security reasons, all updates have to be released according to a very strict protocol: The protocol typically

require a visit to the responsible health center or clinic where the programmer is maintained so that a qualified person can then change the program and recover the prior program. This visit is completed with a writing and signature of various certification documents, etc. These procedures are burdensome and costly. As a result, one typically avoids to follow them too often (for example, no update may be made in the case of a minor modification of the software).

In addition, the security procedures, as complete as they are, do not eliminate totally the risk of an absence of an appropriate update. For example, a programmer that could not be located by the manufacturer will not be updated. Nor do the procedures avoid the risks of operating during the update transition periods, and, especially, the less appropriate utilization of an obsolete version of the software. The reference to "obsolete" should be understood to mean simply a prior version that has been made obsolete by the update being implemented.

Objects and Summary of the Invention

One of the objects of the present invention is, therefore, to remedy to these difficulties, by proposing a new concept of programmer whose software can be updated in an entirely automatic manner, without human intervention, thereby reducing, if not eliminating the burden, costs, and risks associated with current update procedures.

The invention relies on an implementation of the notion of an "object" in the software sense of the term, that is to say a structural element in the broadest sense of the term (data, constructors, destructors and methods), allowing the software to manage and to display data in a very structured manner.

In particular, the access to data is released in general only by methods appropriate to the object. This allows to protect the object data against exceeding a predetermined range and to render the software particularly reliable.

The concept of "heritage" or "inheritance" also facilitates the design of the software, with one object being able to inherit another object. This allows the one object to use directly the methods of the other object and simplify the structure of each with respect to updating by modification or addition of the one object, as will be explained. Examples of such software objects and their applications to the field of cardiac pacemakers will be described below.

In a general manner, one will be able to refer to the published literature on this subject of use of object programming, for example, the work of Max Bouché, *La démarche objet - Concepts et outils*, "AFNOR, 1994, (The Step Object - Concepts and Tools, AFNOR, 1994), as well as the documentation of typical "object oriented" programming languages such as C++.

Broadly, the invention is directed to (i) storing in the implant object oriented software, allowing it to define the display, the processing, and the programming of its data by an external programmer; (ii) comparing in the programmer a list of

objects that the programmer contains to the list of objects that the implant contains; and (iii) downloading from the implant to the programmer objects in the implant that are missing in the programmer or more recent relative to the version of the object that is in the programmer, thereby to update the software of the programmer.

Proceeding in this manner, which is described in more detail hereafter, provides at least four major advantages as compared to the prior art updating procedures: security, economy, simplification of the programming, and standardization.

Security

All programmers are equipped with a basic minimum software, that is to say, objects allowing it to establish communication with an implant, and to manage connections between objects in the implant and objects in the programmer. Such a basic programmer will therefore be able to read any implant and its objects, even if the implant has been developed well after the programmer and contains objects not previously in the programmer. Consequently, the operator of the programmer no longer will be unable to read and re-program the functions of a recent implant model in accordance with the present invention, if he has a compatible programmer, however old it may be. In addition, each parameter has its proper mode of management, defined by software in the implant, which controls the capture of these parameters by the programmer, and which is defined in a given object. Thus, the programmer will not capture false parameter values because they

are no longer controlled by an object pre-existing in the programmer but rather by a object issued from the implant.

Economy

The expense of updating a programmer is considerably reduced and becomes required only for important equipment changes, (namely new generations of programmers, hardware changes telemetry heads, etc.). One also avoids all of the expense of the classic security procedures that go with the known update changes: manufacture of diskettes and manuals for the update, dispatching a qualified operator to each programmer site, signature of a register certifying the update, recovery of the old diskettes, etc.

Simplification of the Programming

The evolution and updating of the programmer software is performed automatically in a manner that is transparent to the user of the programmer.

Only very general functions will be needed to complete management of the basic objects, and the list of objects furnished will, at the end, be completed by the new or additional objects defined in each new implant.

Standardization

The flexibility of the implementation of the present invention allows the creation of a standard allowing all programmers to read compatibly any implant, regardless of its

model or manufacturer. Each manufacturer will be able to define the graphic display aspect of its programmer, the possibilities of data management, supplementary calculations, etc. in a manner that will maintain a certain originality and product distinctiveness, but, with the basic objects being defined in an identical manner, the reading of a "foreign" implant now becomes perfectly possible.

One can thus envisage the realization of a universal programmer, which until now was excluded not only for reasons of cost and complexity, but also for security reasons (namely, the damages from unforeseen incompatibility risks).

One particular aspect of the invention is directed to a system comprising, on the one hand, an implant of the active implantable medical device type, and, on the other hand, an external programmer, in which:

The programmer has a software composed of an number of software objects;

the implant has a memory containing, on the one hand, parametric data concerning the functioning of the implant. and, on the other hand, a number of software objects necessary for the functioning of a programmer in connection with the aforementioned parametric data of the implant;

a bi-directional connection between the programmer and the implant to exchange information allowing particularly the selective transfer of all or part of the aforementioned parametric data and one or more of the aforementioned objects of the implant to the programmer; and

the programmer has a programming capability to command the downloading from the implant of at least a part of the aforementioned objects contained in the implant memory and to add and/or to substitute these objects to the objects defined in the programmer software.

More preferably, the programmer downloads as much or as many of the implant objects as is necessary to become fully compatible with the implant. As a result the system is able to update the programmers. Preferably, the updating procedure occurs automatically when the programmer is placed in communication with the implant.

Advantageously, the programmer capability for updating operates to:

- establish a list of objects that are found in the implant, more preferably a list which includes the version of the objects found;

- compare the established list to the objects in the programmer software;

- command the downloading from the implant of those objects absent in the programmer software and to add these objects to the programmer software; and/or

- command the downloading from the implant of those object for which the version present in the programmer software is prior to that found in the implant, and to replace therewith the objects in the software.

As a result, when an updated programmer is next in communication with a similar implant, the comparison of objects

in the implant to the programmer finds all of the current versions of the objects in the programmer, and no downloading is needed.

Other aspects of the invention are separately directed, as independent products, on the one hand, to the implant, and on the other hand, to the programmer, respectively implementing the appropriate one of the aforementioned characteristics.

Detailed Description of the Invention

One is going now to describe an example of an implementation in accordance with a preferred embodiment of the invention.

As previously indicated, the software of the programmer, as well as that of the implant, is organized in object oriented software "objects."

In the case of an implant which is a cardiac pacemaker, one can represent by one object each element (parameter) of the pacemaker that is programmable or simply displayable on the screen of a programmer or printable on a printer.

Thus, a curve giving the cardiac frequency as a function of time can be represented by a particular software object called "CARDIAC FREQUENCY," that inherits from a more general object, which one can call "CURVE," and comprises a sequence of data corresponding to a curve, with its format, its title, a method of display, a method of printing, etc. The term "format" refers to the scales in abscissa (X-axis) and ordinate (Y-axis).

The programmer for its part has an object "CONNECTION ADMINISTRATOR" that can call data corresponding to the cardiac frequency stored in the pacemaker, by applying transfer data protocols, compression/decompression, etc. as necessary. The programmer will then be able to display a window on the programmer screen presenting the cardiac frequency data, with the help of the object CURVE, allowing all presently available display related functions: zooming, moving in the window, measuring the data displayed with the help of cursors, etc. These functions are understood to be performed in the conventional manner.

The concept of the invention is based on three fundamental hypotheses. First, the programmer software is "open," that is to say that it can be modified by receiving and incorporating new objects concerning data, the display of data, the processing of data, and the programming of the implant. These objects can come from the implant (according to the characteristic of the invention), from a diskette or from any other computer support in itself known (e.g., ROM change, a modem or file transfer), for example, by the preliminary loading of the CONNECTION ADMINISTRATOR object.

Second, the implant stores all objects needed by it to define the programmer display, the data processing, and the programming of data contained in or acquired by the implant. It should be noted that the size of the objects in the implant software can be relatively reduced, to the extent that they inherit from other objects already present in the programmer.

For example, the programmer object CURVE generally defines the display and management of a curve, and therefore the implant software object CARDIAC FREQUENCY need only map the particular features of the data to be displayed to the object CURVE. The object CURVE then manages the actual processing and/or display.

Third, the versions of objects are dated (for example, by a version number and/or a date code) and any more recent version replaces a prior version in the programmer. Developments of objects is thus to be released by imposing an "upward" compatibility, that is to say, the more recent version remains compatible with all data exchanges relative to older versions.

In the beginning of the exchange of data between an implant and a programmer, the programmer systematically calls a block called List-Of-Objects that is a repertory of available objects in the implant. This can be done automatically, preferably as soon as a valid communication link is established. This List-Of-Objects is maintained in the implant. Preferably, the List-Of-Objects also includes the version of each object in the implant. More preferably, the List-Of-Objects includes a table containing an object number and object name (and optionally a description).

The programmer then compares the codes of these implant objects with the objects already known (e.g., an available list of objects in the programmer software, which is preferably dynamically managed and safeguarded in a configuration file). If one or more of these objects is unknown to the programmer, it calls the object CONNECTION ADMINISTRATOR and commences a procedure of downloading the unknown objects so as to complete

its software and to adapt the programmer to the implant. The programmer configuration file is then updated.

The procedure of downloading and updating automatically occurs for each unknown object, by the programmer utilizing the CONNECTION ADMINISTRATOR object that inherits the object being downloaded. The CONNECTION ADMINISTRATOR sends a code, e.g., a number of the object, to the pacemaker and the pacemaker returns a block of corresponding information to the programmer.

A similar procedure is performed for objects that, although present in the software of the programmer, correspond to a version that is older than the one contained in the implant.

After this updating phase, the programmer has all of the objects needed to be compatible with the implant, and can call all data contained in the implant, to display the desired results, re-program the implant, etc. in the conventional manner.

Thus, in the case of an implant having the object CARDIAC-FREQUENCY being read by a programmer not knowing that object, the programmer identifies the object in the List-Of-Objects provided by the pacemaker. The programmer then asks the implant to transfer the object, which it does, and the programmer adds it to its program. The call to the constructor of the object also allows to reserve a memory zone to receive later the curve of frequency, the selection of a display window and the initialization of various variables, associated with the object.

The code (or number) of the object CARDIAC-FREQUENCY then comes to be added to the dynamic list of available objects maintained in the software of the programmer.

When the programmer operator wants to display the curve, it calls on the programmer the corresponding window. The program then tests the presence of the curve in memory, and, if no curve exists, it commences a reading of the cardiac frequency data stored in the pacemaker. The curve is then displayed in a window having the classic display adjustment controls (gain, zoom, moving), measure (cursors) and possibly printing these various functions all being managed by the general object CURVE already present in the programmer.

One is going now to describe two detailed examples of pacemaker objects: an object NOMINAL FREQUENCY (programmed fixed value); and an object RESPIRATORY FREQUENCY (data which varies in the course of the time).

The description of objects can be released under any of several forms, which can be summarized as three cases: (1) machine code, (2) precompiled code and (3) metalanguage: The machine code or firmware is linked to the processor to allow a long term usage. The precompiled code (such as the old fashioned "P-code") is extremely effective, being able to describe all data and methods of the new objects. However, it remains large in size, and is thus less adapted to be used with implants having low memory capacity. The metalanguage specifies only the value of variables belonging to parent objects that are predetermined by the programmer. The volume of data is thereby extremely reduced as compared to the other cases.

Most of data of an implant can be contained in a description in the form of a metalanguage, but the recourse to a precompiled code should remain accessible for the integration with original objects.

In the examples described here, the chosen language is a metalanguage, which gives a reduced size of the objects and a total independence from the programming language. On the other hand, it can only make reference to the basic objects of the programmer, without being able to make them directly progress.

The limits are reached only when a completely new concept will not be able to be integrated by the programmer, with the result that the development of the basic objects has to be extremely well released to avoid the too frequent occurrence of this type of incident that would impose a traditional software update of programmers.

EXAMPLE 1

Object: NOMINAL FREQUENCY

The nominal frequency (also known as a base frequency), which is an essential parameter of all cardiac pacemakers, can be represented by an object described in the following manner:

1. object number:

a unique code is associated with the object, for example, eight characters (two letters for the manufacturer, four letters for the object and two digits for the number of version). This allows to choose a code both unique and easily decipherable.

2. object ANCESTOR:

a number for this object provides the code of the object ANCESTOR, that is the only object that already resides in this case in the programmer and allows the programmer to manage all processes of display, capture, printing and storing of a numerical value. This object inherits, of course, at a minimum, a more general object that is the CONNECTION ADMINISTRATOR. As it concerns a programming by predetermined steps, the values are coded in number of steps. This datum occupies five characters.

The next data provides the parameters of the object ANCESTOR:

(a) descriptive sentence (100 characters) accompanying the numerical value at the display (for example, twenty characters of text for a total of a hundred characters for twenty characters in five different languages).

(b) unit of measure (10 characters).

(c) values of the minimum, the maximum and the nominal value (16 characters): the limit is resident in the object ANCESTOR so as to prevent the capture of a numerical value out of the limits. These limits are updated by the maximum and the minimum and the nominal values replace the values previously present in the object during a call of a method of "placement to the nominal" which inherit all programmable objects. The step increment can be defined as a constant, logarithmic, or inverse function value in order to accommodate to the various cases existing in pacemakers.

(d) - reference window (5 characters): Each datum to be displayed in one of the screens of the programmer, that can contain one or more windows. Each window is defined by the pacemaker with at least a title, a format (one column, two columns, etc.) and actions that one can execute (displacement, call to a momentary reading or continuous reading of the pacemaker, local menu, etc.)

(e) position in the reference window (5 characters): This provides column, height, etc., that can summarize in the form of a number to five digits: The first gives the column, the three following give an order of priority in relation to the others; the nucleus of screen-management places them from high to low by order of priority.

The totality of the data for this object occupy therefore 56 characters in monolingual mode and 136 characters in a multilingual mode.

EXAMPLE 2

Object: RESPIRATORY FREQUENCY

In the case of a pacemaker analyzing the respiratory frequency, the acquired data can be stored for periods of several hours, several days and even several months.

The object can be described in metalanguage in the following manner (Those terms already described are not repeated except as noted):

- (1) number of object (8 characters)
- (2) object ANCESTOR (5 characters) It concerns a curve generator in frequency as a function of time $f(t)$ possessing a vertical scale and a horizontal scale, a format of data and a

display mode (line, minimum - maximum, etc.), methods of scale change, display and printing. This object belongs to the objects residing in the programmer.

The next data give the parameters of the object ANCESTOR:

- (a) descriptive sentence (20 characters for each language, or 100 characters in 5 language multilingual mode):
- (b) unit of measure (10 characters).
- (c) format of data (5 characters): continuation of numerical values (an integer) (8, 16, 24 or 32 bits), real, etc. This format is a resident object that inherits the object CONNECTION ADMINISTRATOR so as to render it capable to manage the transfer of data between the pacemaker and the programmer.
- (d) time period separating each datum, (4 characters): period in milliseconds, coded by a four byte word allowing thus a dynamic range of time of between 1 ms and 1137 hours.
- (e) vertical scale description (12 characters): low value, high value, step increment.
- (f) reference window (5 characters): same as in the case of the nominal frequency.
- (g) position in the reference window (5 characters): as for the nominal frequency, the display of the graph positions according to a certain order of priority; the number of columns and the number of displayed elements are taken into account by the window to grant to each the available maximum place: thus, a window with one

column containing only one curve displays the former on all the available surface, a window with two columns and four curves displays each on a quarter of the total surface, area etc.

The totality of these data occupies therefore 74 characters in monolingual mode and 154 characters in a five language multilingual mode.

One can thus describe in the same general manner all of the necessary parameters for the functioning of the pacemaker.

In the case of a relatively complicated cardiac pacemaker, comprising for example 20 curves stored in memory and 50 programmable parameters, the total size of objects is thus, in multilingual mode, $50 \times 136 = 6800$ characters for parameters and $20 \times 154 = 3080$ characters for curves, for a total of 9880 characters. To this it is necessary to add the block List-of-Objects that will be transmitted to the programmer beforehand to perform the update, which is $70 \times 8 = 560$ bytes.

The total size reserved to the storing of objects in the pacemaker reaches, therefore, approximately 10Kb in the case of the most complex (for example, a DDD rate responsive pacemaker of the most recent generation). Advantageously, these elements can be easily introduced in a ROM circuit of the pacemaker, without entailing over-consumption, so that this circuit is not activated.

One skilled in the art will appreciate that the present invention can be practiced by other than the described

embodiments, which are presented for purposes of illustration and not of limitation.

1. Abstract

A system and methods for the automatic update of the software of an external programmer implant is that is used to program and configure an active implantable medical device implant and acquire data obtained by the implant. The programmer comprises software composed of an assembly of software objects. The implant comprises a memory containing parametric data for the functioning of the implant and an assembly of software objects necessary for the functioning of the programmer in connection with the aforementioned of parametric data. The programmer and the implant communicate bi-directionally. The automatic updating preferably occurs by the programmer reading of the memory of the implant, establishing a list of objects that are found in the implant with their respective versions; comparing the list to the objects (and their versions) that are in the programmer software, downloading from the implant of those objects which are not found in the programmer software and adding these objects to the programmer software; and/or downloading from the implant those objects in the programmer software whose version is prior to the version of the object found in the implant, and to replace the programmer software objects with the more current version.

2. Representative Drawing

None

THIS PAGE BLANK (USPTO)